



Australian Government

BEST AVAILABLE COPY

Patent Office
Canberra

I, MICHELLE HENKEL, TEAM LEADER EXAMINATION SUPPORT AND SALES hereby certify that annexed is a true copy of the Provisional specification in connection with Application No. 2002952711 for a patent by TYPEFI SYSTEMS PTY LTD as filed on 18 November 2002.

WITNESS my hand this
Tenth day of April 2006

A handwritten signature in cursive script that reads "Michelle Henkel".

MICHELLE HENKEL
TEAM LEADER EXAMINATION
SUPPORT AND SALES

CERTIFIED COPY OF
PRIORITY DOCUMENT



Our Ref: 7755170

P/00/009
Regulation 3:2

AUSTRALIA

Patents Act 1990

PROVISIONAL SPECIFICATION

Applicant(s): Typefi Systems Pty Ltd
 Innovation Centre
 90 Sippy Downs Drive
 Sippy Downs Queensland 4556
 Australia

Address for Service: DAVIES COLLISON CAVE
 Patent & Trade Mark Attorneys
 Level 10, 10 Barrack Street
 SYDNEY NSW 2000

Invention Title: **A method of formatting documents**

The invention is described in the following statement:

A METHOD OF FORMATTING DOCUMENTS

The present invention relates to an automated method of laying out page elements for inclusion in a work for printing or for on-line display.

5

PRIOR ART

Current typesetting and document layout systems and techniques utilise different sources of information to produce a completed work. The information content is generally produced separately from the graphical and stylistic content which gives
10 the finished work a particular style. The style may be common to a group of works across a series, lending the series a consistent appearance, designed to appeal to potential purchasers.

The producer of the information content, hereinafter called the author, writes the
15 text of the work, and if required, produces drawings and other graphical figures. The raw text and other material is hereinafter termed the manuscript, and is not formatted in any particular style.

The stylistic appearance is generally controlled by a graphic designer. The graphic
20 designer typically prepares sample pages and produces written guidelines which dictate the finished appearance of the work. The sample pages and guidelines may be created using a known desktop publishing software package such as Adobe PageMaker, Adobe InDesign or Quark XPress. The stylistic information is hereinafter called the book design, and includes the more aesthetic content which
25 controls the appearance of the finished work.

The book design generally includes several different parts: .

Paragraph styles: These are applied to paragraphs within the manuscript and
30 specify information such as the fonts and font sizes to be applied to various elements within the book including the main body text, section headings, sidebar

headers, sidebar text, captions, running headers and lists. Paragraph styles deal only with the format of the paragraph. They do not provide any guidance on the relative or absolute positioning of paragraphs.

- 5 Master pages: These are pages that are used as the basis for all other pages within a document. Typically master pages include elements whose positions and characteristics rarely if ever change throughout one document, allowing these pages to be predefined. Master pages may include background graphics used on part title pages, running heads used on the main text pages, background shading
- 10 behind page margins and placeholders for things such as page numbers and chapter titles. Master pages are also included for the main text pages with a placeholder frame to indicate the position of the book's primary text. Master pages are included by the book designer for every significant page type including Part and Chapter title pages, TOC (Table of Content) pages, and Index pages.

15

- Elements: These are items that change in terms of both position and content. They are defined by the book designer, and may be illustrated with sample text and images (in the case of Figures, for example), and they may have associated positioning rules such as "always place at the top of the page" described in a
- 20 related text document. Elements include sidebars, tables, figures and other items relevant to the book's purpose and design.

- The book design is normally prepared separately from the manuscript, and is submitted to the publisher together with a sample chapter or several sample
- 25 pages, indicating how the finished work will appear when published.

- Once the book design has been approved by the publisher, and the manuscript has been completed, both are sent by the publisher to a typesetter who prepares the finished document by manually combining the manuscript with the design and
- 30 layout rules included in the book design. The process is a manually intensive one,

with scope for error and misunderstanding. A typical work such as a reference book containing several hundred pages may have a fairly complex layout including sidebars, drawings, photographs, graphs and tables, and may take a typesetter from several weeks to several months to prepare manually.

5

The process is very subjective, and even by using a number of positioning rules which define how the positions of certain objects interrelate, it is likely that two different typesetters working independently, but with the same material would produce two very different results.

10

Several attempts have been made to automate the typesetting process, including the development of typesetting systems such as Tex, LaTeX and Advent 3B2 that provide extensive programmatic support for defining automated templates. However, creating a template for a book that will be commercially-attractive can take several months of intensive scripting development. The three desktop publishing packages referred to previously also include scripting systems which allow a certain amount of control over the layout process to be exercised by a suitably skilled programmer. However, they offer a compromise solution at best. A full layout can only be achieved if the intended result is relatively basic. A more complex layout can be achieved, but this requires extensive programming for each new book design, often rendering it uneconomic when compared to the manual process, which it is intended to replace. Typically an automated template is only developed for books whose basic design will be used in many titles, such as in a series of works, where the total title count will number in the dozens or hundreds of examples.

25

SUMMARY OF THE INVENTION

The present invention aims to at least alleviate some of the problems experienced in prior art typesetting systems. In particular, the present invention aims to allow a book design to be simply specified according to a plurality of rules which define positioning of elements within the finished work.

30

There is an increasing tendency for works which would previously have been published in hard-copy format to now only be published as e-books, intended to be viewed on a VDU. Embodiments of the present invention have particular utility in the preparation of printed media and electronic or on-line media that seeks to emulate the look and feel of a printed page. Of course, such e-books may be printed if desired.

In particular, many documents are now presented electronically in Portable Document Format (PDF) as generated and read by Adobe Acrobat TM. This format is intended to preserve a document's layout and format even when viewed on computers which may have different display options and setups. In this way, the creator of a document can ensure that when viewed and/or printed, his intended format is preserved. This is not possible with other online formats, such as HTML, where the display device interprets certain formatting options to achieve a desired effect rather than rendering them in their original absolute form.

Embodiments of the invention are particularly useful in laying out complex publications, such as text books, academic studies, magazines, newspapers, technical journals, marketing reports, statistical analyses and instruction manuals. Embodiments of the invention may also prove useful in the creation of pages for display via the Internet. Such complex publications, especially those forming one of a series of such publications, can be arranged to present the reader with a consistent style which is common to all members of the series. Embodiments of the present invention allow production of publications which conform to defined layouts or styles, with minimal or, at least reduced, manual intervention in the layout process.

Embodiments of the present invention permit pages to be laid out according to a rule, or, more usually, a set of rules, which define desirable page layouts. The rules relate to the positioning of the many different page elements, which make up

each page of the document. The page elements may include textual and graphic elements such as figures, photographs, sidebars, illustration, graphs and tables. The rules are defined in terms that relate a page element either to another page element or a physical property of the page, such as an edge or a margin.

5

Through the application of these rules, using the processes defined in accordance with one or more embodiments of the invention, it is possible to fully automate the layout of pages composed of dynamically-supplied data. For example, a personalised newspaper supplied with data forwarded via a wireless connection
10 such as WAN, Wireless LAN or G3 mobile telephone technologies may be formatted and laid out in such a way on the electronic page or screen as to present to the user a personalised news page that matches current commercial newspaper layouts in terms of style and presentation aesthetics, but allows the user to select particular areas of interest to be presented.

15

In a first broad form, the present invention provides an iterative method of laying out elements on a page for printing or on-line display, wherein the page includes information content and style content, said information content including a plurality of different page elements, and said style content including a rule associated with
20 a particular page element, said rule defining a scoring system which defines a score dependent on a degree of conformance to said rule, the method further including the steps of:

25

- a) arranging on the page, the plurality of page elements included in the information content;
- b) scoring the resulting page layout according to the rule included in the style content;
- c) storing said score; and
- d) repeating the above steps a) to c) for a plurality of different page layouts.

30

The page layout having the best score is preferably selected as the page layout to

- 6 -

be used in the final work. Alternatively, the user may be presented with a selection of the highest scoring layouts and manually selects his preferred layout.

5 Preferably, the method further includes the step of dividing the information content into a plurality of page-sized sections prior to laying out the elements on each page.

Preferably, the page size information is included in the style content.

10 Preferably, every page layout is arranged such that each of the plurality of page layouts includes the plurality of page arrangements in a different arrangement, with each successive layout differing from the previous one in that a particular page element is offset from its previous position by a predetermined distance.

15 Alternatively, and in order to reduce the number of iterations required, each page element is positioned on the page in a position as defined by a rule associated with it. In this way, it is possible to pre-empt the layout process somewhat by estimating which positions are likely to give the best scores, and forcing the elements to occupy those positions which are deemed optimal according to the
20 defined rules.

Preferably, the information content is included in a first computer-readable data file.

25 Preferably, the style content is included in a second computer-readable file.

Preferably, said first and second computer-readable files are created separately.

30 Preferably, certain information from said second computer readable file is available to the first computer readable file. This information preferably includes details of certain defined page elements which may be assigned to certain content

in the first computer readable file.

Preferably, the information content is divided into page-sized portions before the iterative layout process begins. In this way, the layout engine is able to lay out a single page at a time. The division into page sized portions is performed on the basis of the size of the individual page elements making up a page. Each page element identified in the information content is formatted according to the information in the book design data file, and from that process, the size of each page element, such as sidebars or figures, can be calculated and the content of each page determined.

In a second broad form, the present invention includes a system for laying out elements on a page for printing or for on line display, including processing means for receiving a first data file including information content, and laying out the content in the first data file according a rule included in the second data file, wherein said processor is arranged to generate a plurality of different layouts of said information on the page and score each layout according to a scoring scheme included in said second data file.

Preferably, the first and second data files are arranged to be submitted to a remote location which houses the processing means.

Preferably, the first and second data files may be submitted using a suitable data network. A suitable data network is preferably the World Wide Web. Suitable security provisions may preferably be applied to any data transfers to protect any confidential information. Suitable security provisions include the use of Secure Sockets or similar means.

The layout engine may be configured to run on a single computer or server. Alternatively, for added reliability and to provide redundancy, a distributed processing system may be used, whereby the typesetting of a particular work is

split into a number of tasks which may be performed on different machines. A convenient way of splitting the layout task is to pre-process the information content to create a number of smaller sections, such as parts, chapters or pages, each of which may be allocated to a different processor in the distributed network.

5

Preferably, the computer system including the layout engine, which operates to combine the information in the information content file and book design file, is physically remote from the creators of said files. In this way, the publisher who operates the layout engine is able to maintain control of the layout engine. In a particular business model, the publisher distributes the two software packages
10 needed to create intellectual content and style content, and arranges to receive the data files produced and use these to produce the finished typeset work.

In this way, two software packages referred to may be given away for free or for a
15 small price, and the main income may be received for each typeset work which is produced by the layout engine.

Alternatively, and particularly as desktop computers become more powerful, it is possible to integrate all three software processes involved in the layout system
20 (the intellectual content creator, the book design creator and the layout engine) into a single software package which may be operated on a single computer. In this way, an author may create his intellectual content to be stored as a first data file. He may also create a book design, or select one of several pre-defined styles available, and store that as the second data file. The layout process can then be
25 performed locally, with the resultant layout displayed on his screen with no need to contact a remote computer.

BRIEF DESCRIPTION OF THE DRAWINGS

For a better understanding of the present invention and to understand how the
30 same may be brought into effect, the invention will now be described by way of example only, with reference to the appended drawings in which:

Figure 1 shows an overview of the processes, inputs and outputs of an embodiment of the present invention;

5 Figure 2 shows a sample definition of a sidebar element;

Figure 3 shows a completed sidebar including user-supplied content;

Figure 4 shows a further sample definition of a sidebar element;

10

Figure 5 shows a completed sidebar including user-supplied content;

Figure 6 shows a further sample sidebar including graphical content;

15 Figure 7 shows how different features of the sidebar of Figure 6 inter-relate;

Figure 8 shows a sample computer menu used to define a rule;

20 Figures 9a-h show various iterations in an iterative layout process according to an embodiment of the invention;

Figures 10a and 10b show the definition of a particular table style and a sample table produced from said table style;

25 Figure 11a-d show different table style definitions and corresponding sample tables produced from said table styles;

Figures 12a-c show different table style definitions and corresponding sample tables produced from said table styles; and

30

Figures 13a and 13b show a table definition and a sample result of applying said definition to some user entered data.

DETAILED DESCRIPTION OF THE EMBODIMENTS

- 5 Figure 1 shows a top-level view of the configuration of an embodiment of the present invention, and illustrates the data flow between different parts of the system.

- Blocks 100 and 120 represent manual processes. Block 140 represents a process
10 performed automatically. Blocks 110 and 130 represent intermediate data outputs, and block 150 represents the finished work.

- Process 100 involves the creation of the manuscript, or raw information which will form the intellectual content of the finished work. This is typically created by an
15 author who may know nothing of the final layout of the completed work. The intermediate output 110 is a computer readable file including raw text, which may be supplemented in part by some basic markup or tag information such as is used for XML (eXtensible Markup Language). The intermediate output 110 may be stored in a database, enabling it to be reused, at least in part, for multiple titles and
20 in multiple designs.

- Process 120 involves the creation of the book or graphic design. This is typically created by a graphic designer who may know little or nothing about the intellectual content of the finished work. The output 130 of process 120 is a computer
25 readable file which defines how certain elements, which will appear in the completed work, interrelate. The interrelations are defined in terms of rules, which are supplemented with one or more weightings which provide a relative measure of the importance of each rule. The files created by Process 110 may include tagging specifically intended to be included in the design defined in Process 120,
30 or they may contain generic tagging typically conforming to an XML Schema that describes each part of the text and its purpose, such as the chapter heading and

- 11 -

chapter number, the body text, page elements, such as sidebar headers and sidebar text, figures and tables.

- Process 140 is an automated layout process which receives as inputs both the manuscript data file 110 and the book design data file 130. The automated process, as will be described more fully later, lays out the elements defined in the manuscript data file 110 according to the rules defined in the book design data file 130. The layout is performed automatically according to an iterative process which divides the text included in the manuscript data file 110 into page sized sections and lays out each element on that page in a plurality of different ways, each having a slightly different arrangement to the others, and then scores each arrangement according to the rules and weightings defined in the book design data file 130.
- 15 The final output 150 is a data file including a completed work which is in a format suitable for printing or uploading to a suitable WWW server as appropriate. Additionally, there may be provided an opportunity for manual intervention if several alternative layouts have the same or similar page scores.
- 20 Separately from the author, who creates the intellectual content 110 of the work, a graphic designer creates a book design file 130. The graphic designer uses a custom software application (The Designer Application) to create the book design file 130. The Designer Application resembles current Desktop Publishing (DTP) applications, in that it allows the designer to use tools to create different elements and place them in desired positions on a page. The types of elements which may be created in this way include frames filled with fixed text, automated text frames (or placeholders) that may be subsequently extracted from the files 110, fixed image frames, automated image frames (or placeholders), background elements such as watermarks and shading, boxes, lines and all of the other elements that typically may form a book, magazine or other document design.
- 30

The designer is able to define any number of different types of page elements. For instance, he can define elements such as main text pages, chapter title pages, sidebars of different types, tables, figures and Q&A boxes. Each different completed work may include any of these elements, and indeed, new types of element can be created by the designer if a custom element is required for any work.

One of the features of the Designer application which allows the designer such flexibility is the ability to create automated frames. Automated frames are drawn on a blank page using a mouse or cursor control, in a similar way to the way in which frames are created using current DTP systems. However, automated frames differ from known frames in a number of respects. An automated frame is configured to reference a particular paragraph style or a particular XML element tag. Paragraph styles are referenced to the tags attached to the text of the manuscript 110 by the author or, in some cases, by a pre-production editor.

In general, the book design file 130 is created before the manuscript file 110. The arrow linking blocks 100 and 120 indicates that there is a flow of information from the Designer Application to the text creation application. The information passed between them includes details of the defined paragraph styles and page element formats such as the paragraphs or other information that needs to be included in a particular type of sidebar, table or figure. In this way, the author of the manuscript file 110 is able to indicate that certain paragraphs in the text are to be treated in a particular way. However, he does not need to be aware of the overall style of the finished work, merely that he wishes a certain block of text to be placed in a sidebar, for instance.

The manuscript data file creation process 100 may be carried out by the author using a known word processor or text creation computer program. Typical programs for this purpose include Microsoft Word, Word Perfect, or an XML editor. Such programs are familiar to most authors and, as such, as far as they are

concerned, they will notice little difference in the text creation process.

The Word Processing program is provided with an overlay program which allows the author to exercise some control over the final layout of the document. It will
5 generally not allow the author to dictate details regarding the actual position of any page elements, but will allow him to assign certain distinguishing properties to certain elements. For instance, if the author wishes to highlight a paragraph of text which is intended to be placed in a sidebar, i.e. separated from the flow of the main text, and usually boxed, or otherwise distinguished, he may be able to select
10 the text in question and select an appropriate menu option, using mouse or keyboard, to tag the text in question. The options available to the author are determined by the paragraph styles and page elements created in the Designer Application 120 and communicated to the text creation application 100.

15 For instance, when preparing the manuscript, the author may create a short paragraph, with a heading, which he intends to be featured in a sidebar so that it does not interfere with the main text of the work. He is able to select the paragraph heading, and tag it from a menu as 'sbHead', indicating it is to be treated as a sidebar header, and positioned and formatted accordingly. He is also able to
20 select the paragraph text, and tag it as 'sbText'. (The tags 'sbHead' and 'sbText' can be arbitrarily named by the operator of the Designer application. They can also be mapped from an XML schema imported into the Designer and Authoring applications.)

25 The text in question may be tagged in a way that remains normally invisible to the author, other than if he chooses to examine the properties of a particular item of text, or the text may be displayed in a distinctive manner, perhaps in bold, underlined or shown in a different colour. Of course, any combination of these may be used.

30

The tagging of the text may be achieved using a markup language as is known

from HTML and XML, or a machine readable labelling system may be used. In any event, the author is able to clearly and simply delineate certain elements from the main text of the manuscript.

- 5 Once the author has completed his work, and tagged it, if desired, he can forward the completed computer file to the publisher, or directly to the Publishing Engine 140.

Figure 2 shows how a sidebar element 200 may be defined. The sidebar consists
10 of two automated frames: the first 210 including the sidebar header text, or title, and the second 230 including the sidebar text. When creating automated frames, the designer creates each type of element on a separate page using the designer application. Since the final size and absolute position on the page of the completed work is unknown, the only significant dimension of the automated
15 frames 210 and 230 are their widths, which are set to the width defined here. The height of the frames is determined by the amount of text which fills them, and this is determined by the author at process 100.

The formatting, i.e. non-positional features, of the automated frames are defined
20 by the designer in the book design data file. In the example shown in Figure 2, automated frame 210 includes the sidebar header 220, and this is formatted so that the text is left justified and appears in Bold 12pt Arial font. Automated frame 230 includes the sidebar text 240, and this is formatted so that the text is left justified and appears in 12pt non-Bold Arial font. Other properties may be added to
25 each automated frame by the designer as he creates them. Other typical properties may be text colour, background shading for the frame, border style and colour.

Figure 3 shows an example of how a sidebar might appear in the finished work
30 150 when formatted according to the rules in the book design file 130. The title is shown in a separate box to the text, and is presented in a bold font as specified in

the template shown in Figure 2. The text is sourced from the manuscript document 110 where the paragraphs have been linked to the tags or styles named '[sbHead]' and '[sbText]'.

- 5 Another feature of the automated frames is their ability to reference special fields, allowing them to be used to insert an incrementing counter, another part of the text from elsewhere in the book for cross referencing purposes, chapter and title numbers, catalogue numbers, information referenced from a database, or other information or data available in machine-readable form. The designer is able to
- 10 specify exactly what information may be inserted, and from where it is to be sourced.

- Another feature of the automated frames is their ability to extract multiple related paragraphs from a manuscript. For example, the 'sbText' tag or style reference
- 15 may be placed in an automated frame with a 'Repeat' function. The formatting engine 140 uses this option to trigger a behaviour wherein all further paragraphs following sequentially from the first 'sbText' paragraphs that have been assigned the 'sbText' style or tag will be incorporated into the current element. This allows elements with an unknown number of paragraphs to be incorporated into the
- 20 original element design.

- Figure 4 shows how a sidebar template may be created in the designer application which includes the previously described elements of Header and Text with 'sbText'
- 25 incorporating a 'repeat' option described above. It also includes new parts 'Sidebar' which is just plain text reading 'Sidebar', {Ch#} and {Sb#} which are automatically incrementing fields which insert the current chapter number and sidebar within that chapter respectively.

- 30 Figure 5 shows an example of how a sidebar formatted according to the template of Figure 4 may look if it is the third sidebar in the second chapter of the finished

work 150.

Figure 6 shows a typical page element 400, known as a sidebar. Sidebars are often included in books, and generally provide short summaries of topics, interesting facts, illustrative graphics or other text related to the main, or body, text.

The sidebar 400 shown in Figure 6 includes a frame 410, having a dropped shadow in the form of an offset, partially obscured shaded frame 420. Inside the frame 410 is a title 430, which provides some information on the topic of the sidebar. The title 430 is separated from the main body of the sidebar by a horizontal rule 440. Beneath the horizontal rule 440, is the main body of the sidebar 400. In this case, the main body consists of a graphic image 450, although it could alternatively be a text passage, an equation, a graph or any other printable item.

In prior art layout systems and methods, the individual elements described above would generally be created and placed manually, resulting in a time-consuming and labour intensive process. However, embodiments of the present invention permit page elements such as the sidebar 400 to be created automatically according to pre-defined rules.

Figure 7 shows the sidebar of Figure 6 with the addition of several structural elements which facilitate the automation of the layout process. The individual components of the sidebar are structured in a hierarchical relationship, such that the position of any one component is dependent on the position of at least one other component.

In the example of Figure 7, the sidebar is created initially from the border 435 around the title text 430. The border may be created automatically, or placed by a layout operator depending on the system being used. The border defines the size

and position of, and is a placeholder for, the title text 430.

Horizontal ruling 440 which separates title text 430 from the main content 450 of frame 410 is defined as the line joining points 442 and 444, shown as diamonds in Figure 8. The vertical positions of points 442 and 444 are defined in terms of the vertical position of the lower edge of border 435. In effect the vertical position of the line 440 is defined as being equal to the vertical position of the lower edge of border 435 with a 0mm offset. In this way, the line 440 lies exactly on the lower edge of border 435. Of course, the offset can be set to any positive or negative value to achieve a different effect.

In this way, if the lower edge of border 435 is moved, then the line 440 will move in a corresponding manner.

In a similar fashion, the upper edge of border 455, which surrounds the graphic image 450 forming the main content of frame 410, is defined in terms of having a 0mm offset from horizontal ruling 440. In this way, any movement of the title text 430 will result in line 440 moving due to the previously define dependency, and the image 450 moving due to the dependency on line 440. The lower edge of border 455 is defined in terms of the size of the image 450. If the image is changed for another, or re-sized, then the lower border is adjusted automatically as necessary.

The position of frame 410 is dependent on the lower edge of border 455. In the present example, the position is defined with a 0mm offset, although this can be altered to leave a greater margin around the graphic image 450.

Finally, the last hierarchical relationship defined for the sidebar 400 specifies the position of the shadow frame 420. Unlike the other relationships defined this far, the shadow frame 420 is defined in terms of the position of the lower edge of frame 410, plus an offset of 6mm. A similar offset is defined in relation to the right-most edge of frame 410, giving the characteristic offset appearance of the shadow

frame 420.

The various hierarchical dependencies defined in sidebar 400 are illustrated in Figure 7 by double-lined arrows.

5

The effect of the hierarchical dependencies is that if the position of any single component alters, then the position of any component which depends on the altered position, either directly or indirectly, is also altered automatically according to the defined relationships.

10

Another feature of the automated frames is their ability to repeat themselves horizontally across the page and vertically down the page for the purpose of defining and rendering tables. A single frame set with an option to repeat horizontally and vertically can act as the basis for a table comprising multiple columns and rows. This can be seen in Figure 10a which shows how such a frame may be defined. Figure 10a shows the on-screen display as seen by a user of the authoring application 100 which includes the relevant stylistic information from the style information file 130. The ticks shown in the boxes marked 'Repeat Horiz' and 'Repeat Vert' indicate that the corresponding frames are to be repeated as more information for them is provided by the author. The formatting of the lower right hand cell – white text on a black background – is repeated as new data is added.

15
20

Figure 10b shows a view of a sample table as it would appear in the finished work 150 on the basis of the table definition shown in Figure 10a.

25

The repeating ability of a cell defined within the Designer application 120 is re-interpreted by the Authoring application 100 to prompt the manuscript author for the appropriate number of rows and columns. By combining repeating cells and non-repeating cells within a single table definition it is possible to create any format of table with any combination of formatting options, from simple clear

30



shading through to complex alternating vertical and horizontal patterns.

This characteristic is shown in Figure 11a which shows how a table may be defined having alternating shading patterns. The table definition shown in Figure
5 11a forces the two rightmost columns to repeat as data is added to the table.

A table produced using the definition of Figure 11a is shown in Figure 11b, where the alternating light and dark shading defined in Figure 11a can be clearly seen.

10 This type of behaviour can be created in horizontal and vertical directions simultaneously to produce a checker-board effect. Figure 11c shows a table definition where light and dark shading alternate in horizontal and vertical directions. Figure 11d shows a sample table resulting from such a definition.

15 In all the example of Figure 11, the addition by the author of more data in further rows or columns results in the automatic application of the format information defined in the table definitions in the style data file 130.

The repetitive behaviour of a linked item is controlled by the user. For example,
20 the border of a background shadow frame may be linked to the border of a repeating cell within a table. Figure 12a illustrates how a user may define a table to include a number of cells as shown, having a dropped shadow.

If the repetitive behaviour of the cell shown in Figure 12a is set to "on first
25 instance" by the user, through an appropriate menu option, the shadow will appear behind the first cell only, irrespective of the number of cells making up the table.

If the repetitive behaviour is set to repeat "on each instance" of the table cell, a copy of the original shadow will be placed behind every cell making up the table,
30 as shown in the sample table of Figure 12b.

If the background shadow is set to "span", the shadow will be drawn only when the

last cell has been placed in the table and will stretch from the original instance of the table cell to the last instance of the table cell. The type of table resulting from this table definition is shown in Figure 12c, where the background shadow is contiguous and spans all the cells making up the table.

5

Certain fields such as were described earlier in discussing 'automated frames' and 'incrementing counters' can be linked to repeating frames to achieve specific results such as an incrementing counter. For example, a frame containing an incrementing counter may be linked to a table cell and set to repeat each time the
10 table cell appears within the current table. In this way a line counter may appear outside the table, automatically replicating and incrementing itself each time a new line within the table is created.

Figure 13a shows the definition of a single column table that acts as the recipient
15 of callout information for the figure to its left. The incrementing counter (shown as a '1' in Figure 13a) is attached to the table cell with a 'repeat on every' property turned on. This causes it to copy itself beside each table cell as the callouts are processed Figure 13b shows a sample result where specific elements, numbered 1 to 4 in the figure on the left are referenced by corresponding numbers in the
20 table on the right. As more elements in the figure are added and referenced, a corresponding numbered entry in the table is created.

As well as defining different paragraph styles, which include formatting options as well as some basic positional and/or dimensioning, such as width for a sidebar,
25 and replication and resizing properties, the designer application may be used to define a series of rules which are then used by the typesetting process 140 to produce the final work 150.

Figure 6 shows a sample popup menu from the designer application which may be
30 used to define positional rules for each element which will appear in the final work. The menu is presented to the designer when he selects an option to assign rules

to a defined element, such as a sidebar or a graph.

A sample rule may be, as shown in Figure 6, 'Is the object aligned to the bottom of the page?'. This particular property may be desirable for certain page elements.

- 5 The sample rule shown in Figure 6 has four numeric quantities associated with it, The first one, labelled 'Max allowable gap' indicates that the rule will score 30 points (out of a maximum 100) when the associated page element is within the 'Max Distance' – shown as the third quantity – of 5mm of the best possible position, i.e. when it is absolutely level with the bottom of the page. The second
10 quantity – 'Points deducted per mm' – indicates that for every mm that the element is positioned away from the ideal location, 3 points will be deducted from the score. The fourth option indicates that points will no longer be deducted once the element is 10mm from the ideal position.

- 15 The rules are stored in the computer readable design file 130 along with the style definitions, and can be used with a range of different source manuscripts 110. This allows one design file 130 to be used in the typesetting 140 of any number of books which may form a consistent series. For instance, a series of books produced by a single academic text book publisher can all be typeset using the
20 same design file 130, resulting in a whole series of works 150 which conform to a house style, with no need to manually prepare the works from scratch each time.

- Other rules may be defined as necessary. For example, a rule may be defined to check if an object is on the same page as its anchor, with 50 points being awarded
25 if it is, and 0 points if it isn't. This rule shows a simple binary rule which either scores maximum points if a condition is satisfied, and nothing if it is not. This is in contrast to the earlier rule described which allows for some deviation from the ideal position, but penalises that particular layout accordingly.

- 30 The final step in the book production process is the typesetting operation 140. The inputs to this automated process are the manuscript file 110 and the book design

file 130.

The layout process is performed using a dedicated computer operating at a physically remote location. The layout process may be configured to run on a single server or computer. Alternatively, the layout engine may include several computers or servers arranged to operate in a distributed manner. This approach offers increased reliability and efficiency, as tasks can be shared between different computers, and in the event that a single machine fails, tasks can be re-allocated as needed.

10

The distributed system may include several distinct computers arranged on a LAN or WAN as needed. Known distributed and load-sharing techniques may be used to manage such a system.

15 Alternatively, the layout engine may be integrated into a single software package configured to run on the same computer as the authoring and book design programs. This approach enables the author to take full control of the authoring, design and layout processes, and does not require data access to a remote computer.

20

The first step in the typesetting process involves extracting the tagged text from the manuscript file 110, and formatting it according to the style information contained in the book design file 130. The formatting extends only to font, character size and insertion of figures, graphs, sidebars and the like. For instance, sidebars are created by extracting the tagged text and formatting the text according to the definitions in the book design file. In this way, the overall size of the sidebar is determined based on the amount of text to be included and the width which was fixed in the book design file.

30 After all text and other material has been extracted and formatted so that it can be sized, the next step is to divide all the material into page sized sections. The page

size and other formatting information is fixed in the book design file 130.

Using a recursive process, the layout engine 140 generates a plurality of different layouts according to which elements are present on each page. If a particular page consists of only body-text, then there is generally only one format possible, as the body text simply fills the space available. However, if a page has any content which is non-body-text, such as elements that may suit a number of different positions on the page, then the layout engine iteratively arranges the page elements in different positions on the page, scores each layout according to the rules applied to each element, and elects to use the highest scoring arrangement in the final layout.

The iterative process is illustrated in Figures 9a – h. The page area available for layout is represented by box 330. The area outside of this is reserved for margins, headers, footers or page numbering, and none of the content of the manuscript file is included there.

This particular page includes a graphic 300, a sidebar 310 and an item of framed text 320, as well as body text which can be arranged to fill the remaining space on the page. The dotted horizontal lines indicate the minimum increment 340 by which the position of the various page elements can be altered. The dimension of the increment 340 is exaggerated in the figures, and is likely to be set to approximately 1mm in practice.

Figure 9a shows the initial layout of the elements on the page after the entire manuscript file has been sized. In between and around elements 300, 310 and 320 runs the body text which has been positioned on that page. The typesetting process 140 evaluates the rules associated with each element on the page, including any associated with the body text, which is not shown in this representation, and stores the result for that page layout.

The next steps involve re-arranging the various page elements into a plurality of possible positions, whilst retaining the same order of appearance on the page. Figure 9b shows that the next layout to be evaluated involves boxed test 320 being moved one increment down the page while the other elements remain as they were in Figure 9a. This new layout is evaluated according to the same rules as before and the new result is stored for this layout.

This process of moving element 320 downwards one increment at a time is repeated, and the rules for the page evaluated each time, until the element 320 reaches the lowest point it can occupy on the page, as shown in Figure 9c. At this point, after storing this page's score, element 310 is now moved down the page by one increment, and element 320 moves back up the page to be positioned just below element 310. This is illustrated in Figure 9d.

Again, the page is scored according to the rules, and the process of shifting element 320 down the page one increment at a time and scoring each layout continues until, again, element 320 reaches the lowest possible position on the page as shown in Figure 9e.

Figure 9f shows the next step immediately following that shown in Figure 9e. Element 310 moves down a further increment, and element 320 moves to be just below element 310. The entire scoring and shifting process continues until both elements 310 and 320 are positioned as low as they can be on the page. At this point, element 300 is shifted down 1 increment, as shown in Figure 9g, and the entire process repeats.

The final step in the iterative process for this particular page is shown at Figure 9h, where all three elements 300, 310 and 320 are positioned as low down the page as they can be. The final score for the page is then stored.

All the scores which have been stored for the possible layouts of the page are

- 25 -

stored and referenced to the particular layout which produced that score. The typesetting process is configured to search through all the stored scores, of which there may be several thousand, and determine which layout produced the highest score, and thus is considered the most suitable layout according to the rules defined in the book design file 130. The highest scoring layout is thus chosen as the layout for that page. The page is configured according to the chosen layout and copied to the master document for eventual output from the process. The typesetting process then moves on to the next page, and the iterative process begins again for the new page. The entire iterative process is repeated for every page in the work.

In the embodiment described above, every possible placement as defined by the minimum placement distance 340 is evaluated against the rules. This can result in an enormous number of calculations being required before a particular layout is chosen. Depending on the computational facilities available, the above described embodiment offers an exhaustive process to determine the optimum layout for a given page. However, in practical terms, the vast majority of possible layouts created using such a scheme will score very poorly and so be rejected.

In practice, the layouts producing the best scores are those where the page elements are positioned closest to their optimal positions as defined in the associated rules. In another embodiment, therefore, the iterative layout process is constrained somewhat, compared to the previously described embodiments.

As an example of the constrained process, consider a page including six distinct page elements, A, B, C, D, E and F. In total, in this example, there are four possible rules which can be used to define the position of each element. The rules are:

1. Set element next to element reference anchor
2. Set element at top of page

3. Set element at bottom of page
4. Keep element on same or later page as element reference anchor.

These rules are exemplary, and other rules may be defined. The rules are each associated with a scoring regime as described previously so that exact conformance with a rule will produce a better score than only partial conformance.

In order to limit the number of iterations performed, the possible positions for the six page elements are defined by the four different rules.

10

In this way, the first iteration attempts to place all six elements according to the first rule. In most cases, it will not be possible to place all elements in the position dictated by a single rule, so certain elements will score well, and others will score poorly.

15

The second iteration attempts to place the first five elements according to the first rule, and the final element according to the second rule. Again, this layout is scored and stored.

20 The third iteration attempts to place the first five elements according to the first rule, and the final element according to the third rule. This layout is scored and stored.

The table below shows the possible layouts which are attempted in this particular example. The iterations are shown in the left hand column, while the rule applied to each page element is shown in the main body of the table. Iteration 1 therefore shows that each element, A-F is placed according to rule 1. At each new iteration, one or more page elements is re-positioned according to a new rule until the final iteration is reached when all elements have been placed according to every possible combination of rules.

30

- 27 -

Iteration	Page Element					
	A	B	C	D	E	F
1	1	1	1	1	1	1
2	1	1	1	1	1	2
3	1	1	1	1	1	3
4	1	1	1	1	1	4
5	1	1	1	1	2	1
6	1	1	1	1	2	2
7	1	1	1	1	2	3
8	1	1	1	1	2	4
9	1	1	1	1	3	1
...
4090	4	4	4	4	3	2
4091	4	4	4	4	3	3
4092	4	4	4	4	3	4
4093	4	4	4	4	4	1
4094	4	4	4	4	4	2
4095	4	4	4	4	4	3
4096	4	4	4	4	4	4

In a page having m elements and n possible rules, then there are n^m possible combinations to be attempted, scored and stored. In the present example which has 6 page elements and 4 rules, then there are 4^6 , or 4096, combinations to attempt.

This number of possible layouts, although it requires a large number of iterations, is still significantly lower than the number required using the previous embodiment.

Once each iteration has been performed, the layout software is able to select the

highest scoring page for presentation to the user. Alternatively, a selection of the highest scoring pages or those scoring above a predetermined minimum may be selected to allow the user to choose which layout is to be used in the final document.

5

An additional rule which has a particular relevance to the above described method can be defined which forces all elements on a particular page to be retained in the same order in which they appear in the manuscript data file 110. In this way, a large number of possible iteration may be easily discarded if the resultant layout breaks that particular rule. An example of where this rule may be used is in the case of a heading and a sub-heading, where the sub-heading has to follow the heading, and so any layout which places the subheading before the heading can be scored as a zero and discarded with no further evaluation being required.

10

15 Before the finished work 150 is created, a post production process is required. This process performs formatting which can not be completed until the main layout is finalised, and includes the addition of page numbering, running heads, cross references and table of contents creation. The creation of a table of contents, for instance, requires each page to be formatted and numbered.

20

The typesetting process 140 is arranged to be performed using a suitably programmed computer. The program accepts as inputs the manuscript data file 100 and the book design file 130. The program is arranged to process the data from each file in the manner already described, and to produce as an output, a computer file in a format suitable for printing. A suitable format is Adobe Portable Document Format (PDF). Other formats may be suitable, and the program may be configured to allow a user to select alternative formats as desired.

25

The typesetting process is intended to be largely free of user intervention, and performs the layout task primarily on the basis of the rules defined in the book design file 130. However, there may be occasions when human intervention is

30

desirable. For instance, two or more layouts for a particular page may produce equal or similar scores, which may require a human operator to select one option only.

5 In a preferred embodiment, the typesetting process 140 is performed remotely from the text creation 100 and Designer 120 applications. In this way, the book designer and author can perform their work largely independently of each other, and submit their respective files via the Internet, for instance, to a publishing house which houses the computers which perform the typesetting process. The
10 typesetting process can then be completed on the basis of the two submitted files, and a draft copy of the finished work 150 can be supplied to the author and/or book designer automatically as soon as the process 140 is complete. The whole typesetting process from receipt of files 110 and 130 to dispatch of the finished work 150 can be completed in a matter of minutes, rather than the days or weeks
15 presently needed to typeset a work.

In many cases, the book design file is available before the intellectual content of the finished work. In such a case, an author may elect to view a preview of a chapter or the whole work. To do this, he may select an appropriate option from a
20 menu of the authoring program, which sends the current chapter or work to the layout engine via a suitable data link, such as the Internet, together with a reference to the associated book design file which may already be stored with the publisher. The layout process is then able to layout the submitted content according to the existing book design file. The typeset work is then sent back to
25 the author in a suitable format for display. A preferred format is PDF.

In the event that a specific book design file is not available, the author may select one of a number of predefined styles which may be made available by the publisher. Indeed, in many cases, one of these predefined styles may be suitable
30 for the finished work, particularly in less complicated works.

- 30 -

The present invention includes any novel feature or combination of features disclosed herein either explicitly or any generalisation thereof irrespective of whether or not it relates to the claimed invention or mitigates any or all of the problems addressed.

Dated this 18th day of November, 2002

TYPEFI SYSTEMS PTY LTD

By Its Patent Attorneys

DAVIES COLLISON CAVE

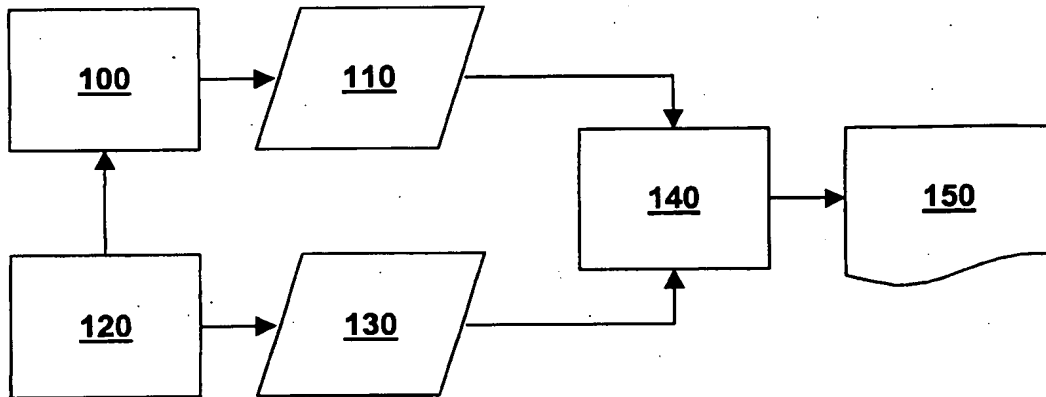


FIGURE 1

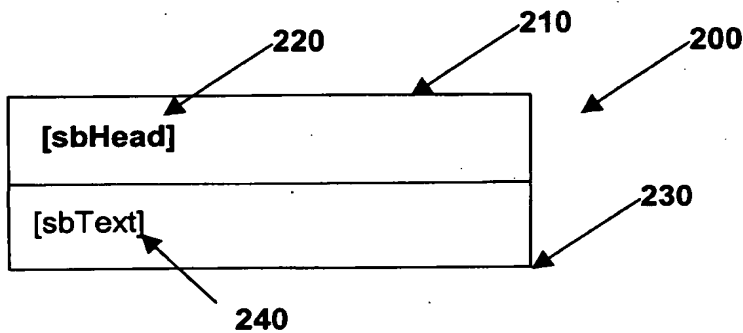


FIGURE 2

Automatic File Opening

You can set Windows to automatically open your Accounting data file when you launch the software. For details, see *Automatically Opening Your Data at Start-up* later in this chapter.

FIGURE 3

Sidebar {Ch#}.{Sb#}: [sbHead] – [sbText]|r

FIGURE 4

Sidebar 2.3: Automatic File Opening – You can set Windows to automatically open your accounting data file when you launch the software. For details, see *Automatically Opening Your Data at Start-up* later in this chapter.

FIGURE 5

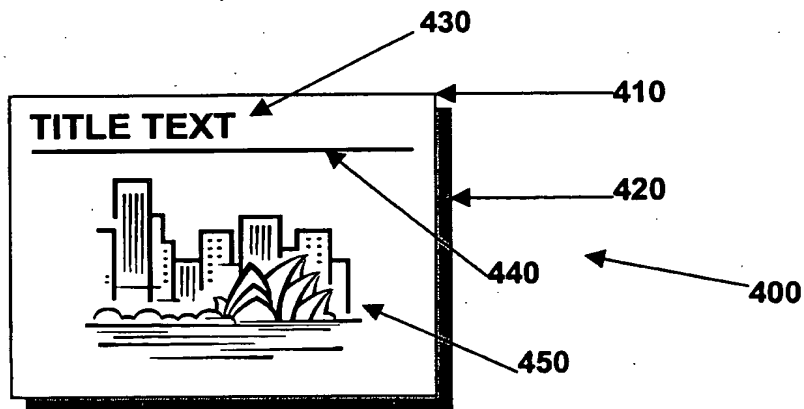


FIGURE 6

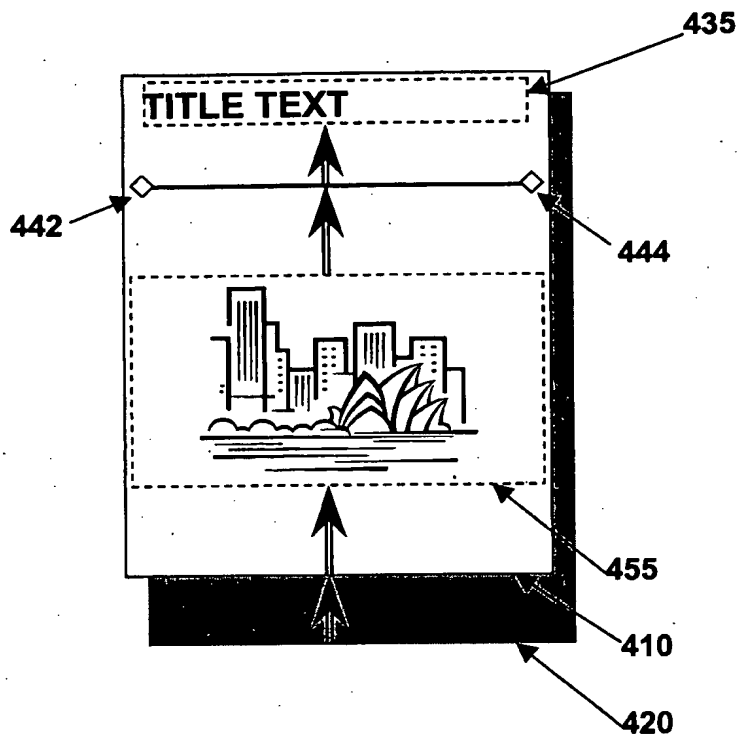


FIGURE 7

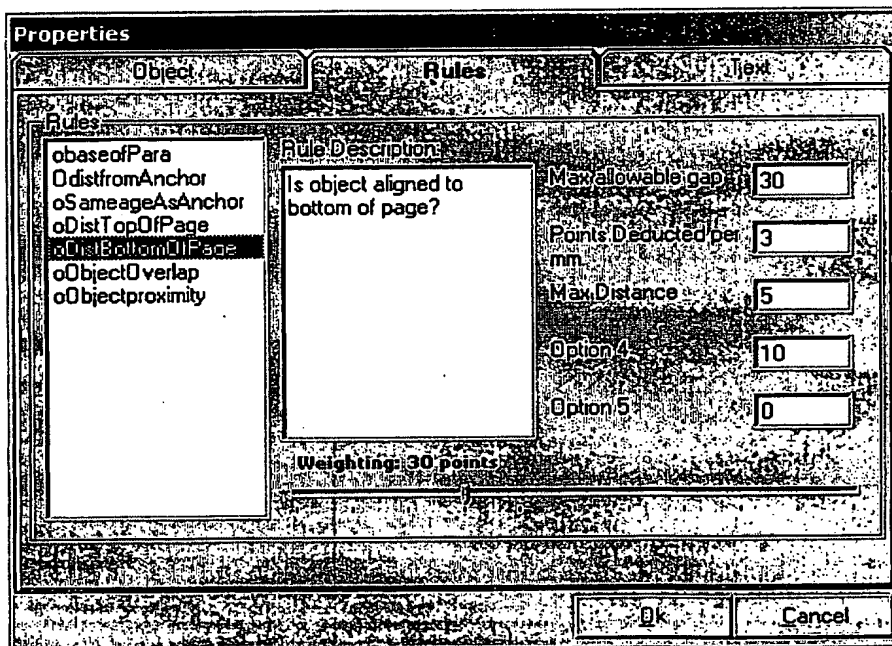


FIGURE 8

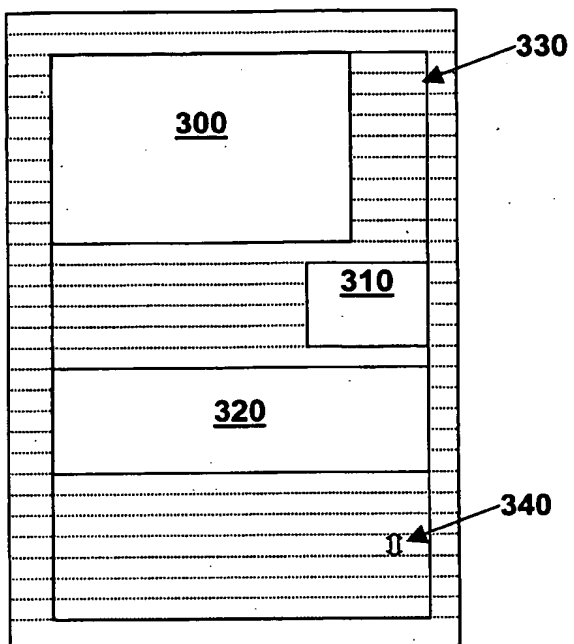


FIGURE 9a

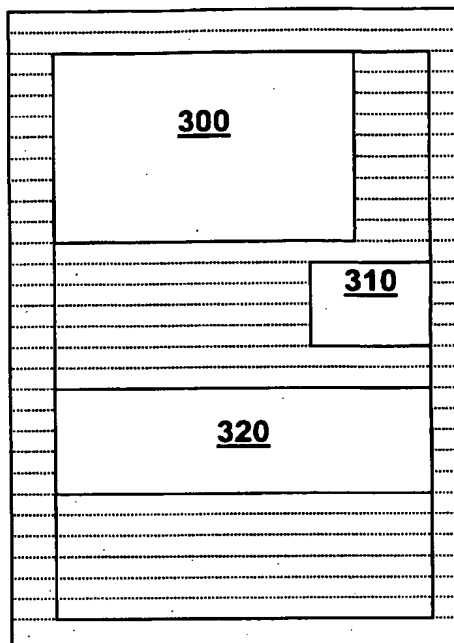


FIGURE 9b

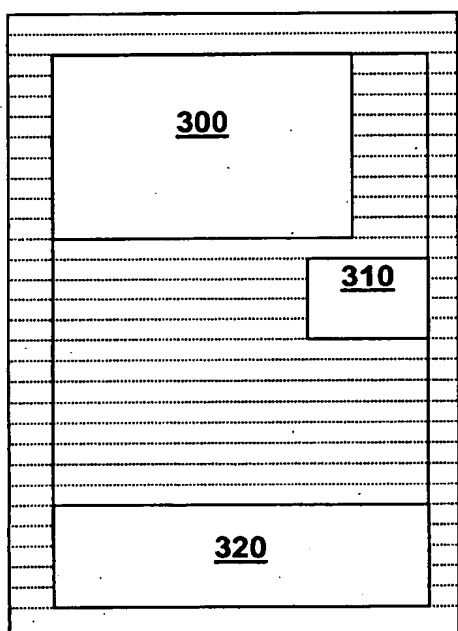


FIGURE 9c

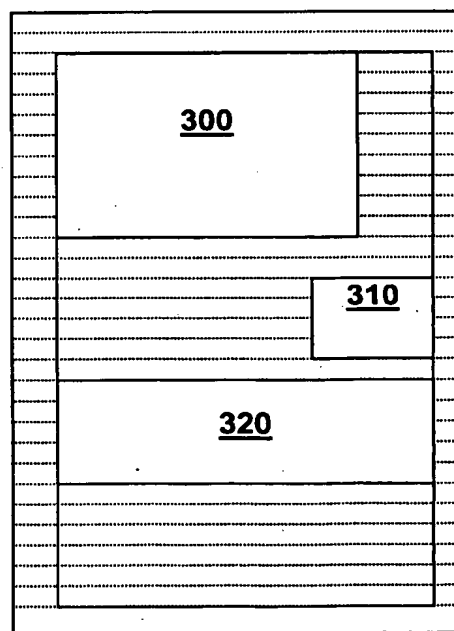


FIGURE 9d

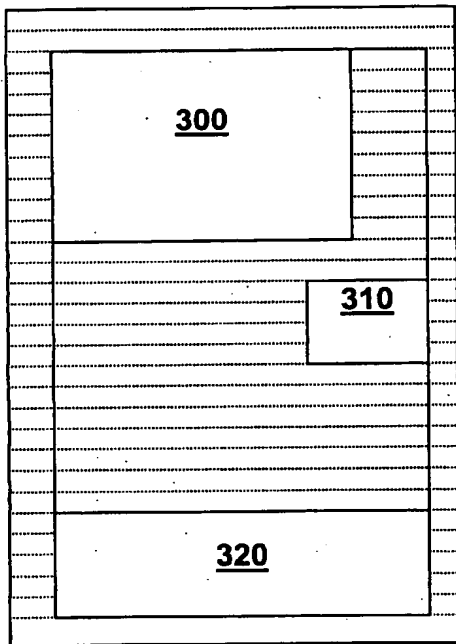


FIGURE 9e

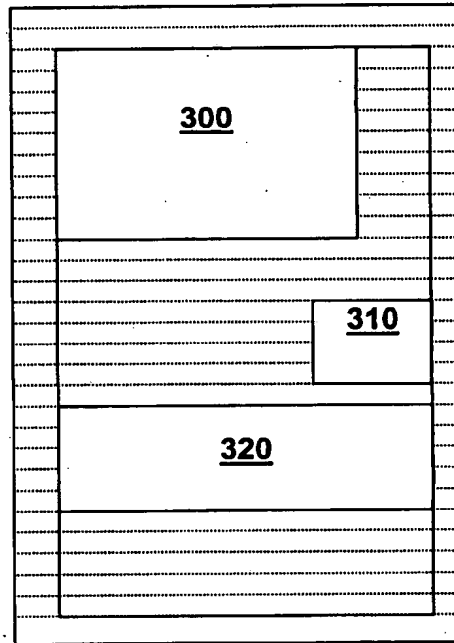


FIGURE 9f

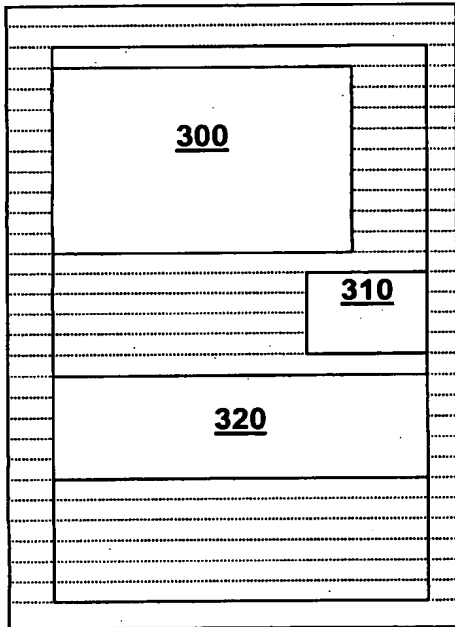


FIGURE 9g

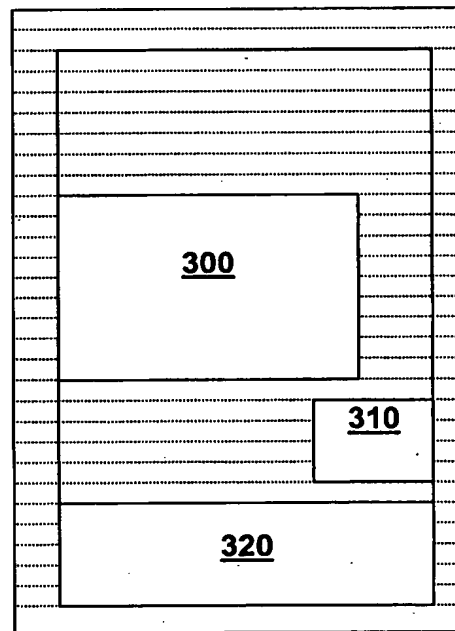


FIGURE 9h

☐ Repeat Horiz ☒ Repeat Horiz
 Repeat Vert ☐ TOTAL SALES Period
 Repeat Vert ☒ [Region] [Amount]

In this table the second column is set to repeat horizontally and the second row set to repeat vertically. The number of repeats is driven by the number of rows and columns in the submitted manuscript.

FIGURE 10a

TOTAL SALES	Q1	Q2	Q3	Q4
East	20,000	30,000	25,500	28,600
South	30,200	38,000	29,500	31,300
North	26,000	25,500	30,000	28,600
West	38,000	25,500	29,500	30,200

This table was created using the format described in FIGURE 10a. It has been automatically expanded to include any number of repeating columns and rows.

FIGURE 10b

☐ Repeat Horiz ☒ Repeat Horiz ☒ Repeat Horiz
 Repeat Vert ☐ TOTAL SALES Period Period
 Repeat Vert ☒ [Region] [Amount] [Amount]

In this table a second repeating column is used to define a repeating alternating pattern. The number of columns drawn is determined by the manuscript data and the second column won't be drawn on the page unless required.

FIGURE 11a

TOTAL SALES	Q1	Q2	Q3	Q4
East	20,000	30,000	25,500	28,600
South	30,200	38,000	29,500	31,300
North	26,000	25,500	30,000	28,600
West	38,000	25,500	29,500	30,200

This table was created using the format described in Fig a. It has been automatically expanded to include any number of repeating columns and rows.

FIGURE 11b

	<input type="checkbox"/> Repeat Horiz	<input checked="" type="checkbox"/> Repeat Horiz	<input checked="" type="checkbox"/> Repeat Horiz
Repeat Ver <input type="checkbox"/>	TOTAL SALES	Period	Period
Repeat Ver <input checked="" type="checkbox"/>	[Region]	[Amount]	[Amount]
Repeat Ver <input checked="" type="checkbox"/>	[Region]	[Amount]	[Amount]

In this table a second repeating row is added to the definition shown in Figure 11a, achieving a more complex alternating pattern. Any number of repeating rows and columns may be defined. It isn't a requirement that each alternating pattern equal a complete instance of that pattern as shown in Figure 11b.

FIGURE 11c

TOTAL SALES	Q1	Q2	Q3
East	20,000	30,000	25,500
South	30,200	38,000	29,500
North	26,000	25,500	30,000
West	38,000	25,500	29,500

This table was created using the format described in Figure 11c and is comprised of alternating columns and rows. The pattern was repeated 1.5x horizontally and 2x vertically.

FIGURE 11d

	<input checked="" type="checkbox"/> Repeat Horiz	
Repeat Ver <input checked="" type="checkbox"/>	[TableData]	Shadow attached to table cell frame using a 'repeat' property.

A simple table definition where a background shadow frame is linked to the table cell.

FIGURE 12a

[TableData]	[TableData]	[TableData]
[TableData]	[TableData]	[TableData]

When linked to repeat 'on every' the background frame appears behind every table cell.

FIGURE 12b

[TableData]	[TableData]	[TableData]
[TableData]	[TableData]	[TableData]

When linked to as a 'span' the background frame stretched from the top-left corner of the first instance of the table cell, to the lower-right corner of the last instance of the table cell.

FIGURE 12c

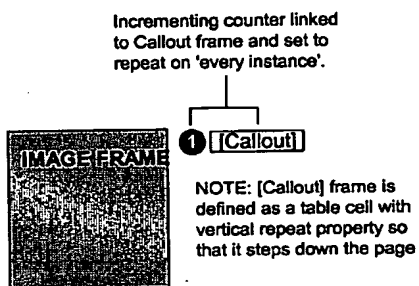
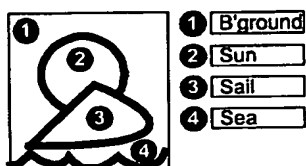


FIGURE 13a



Counter frame replicates and increments on each instance of the callout frame.

FIGURE 13b

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

☒ **BLACK BORDERS**

☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**

☐ **FADED TEXT OR DRAWING**

☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**

☐ **SKEWED/SLANTED IMAGES**

☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**

☐ **GRAY SCALE DOCUMENTS**

☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**

☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**

☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.